Middleware Enabling Science in Distributed Systems

Deb Agarwal
Distributed Systems Department

Lawrence Berkeley National Laboratory

DSD Overview

- Collaborative Interaction Tools
- Grid middleware
- Security
- End-2-End Monitoring
- Transport mechanisms

DSD High-Level Goals

- Allow scientists to address complex and large-scale computing and data analysis problems beyond what is possible today
- Develop software components which will operate in a distributed environment
  - Middleware providing basic services and capabilities in the Grid
  - Applications and middleware supporting synchronous and asynchronous collaboration between geographically remote researchers

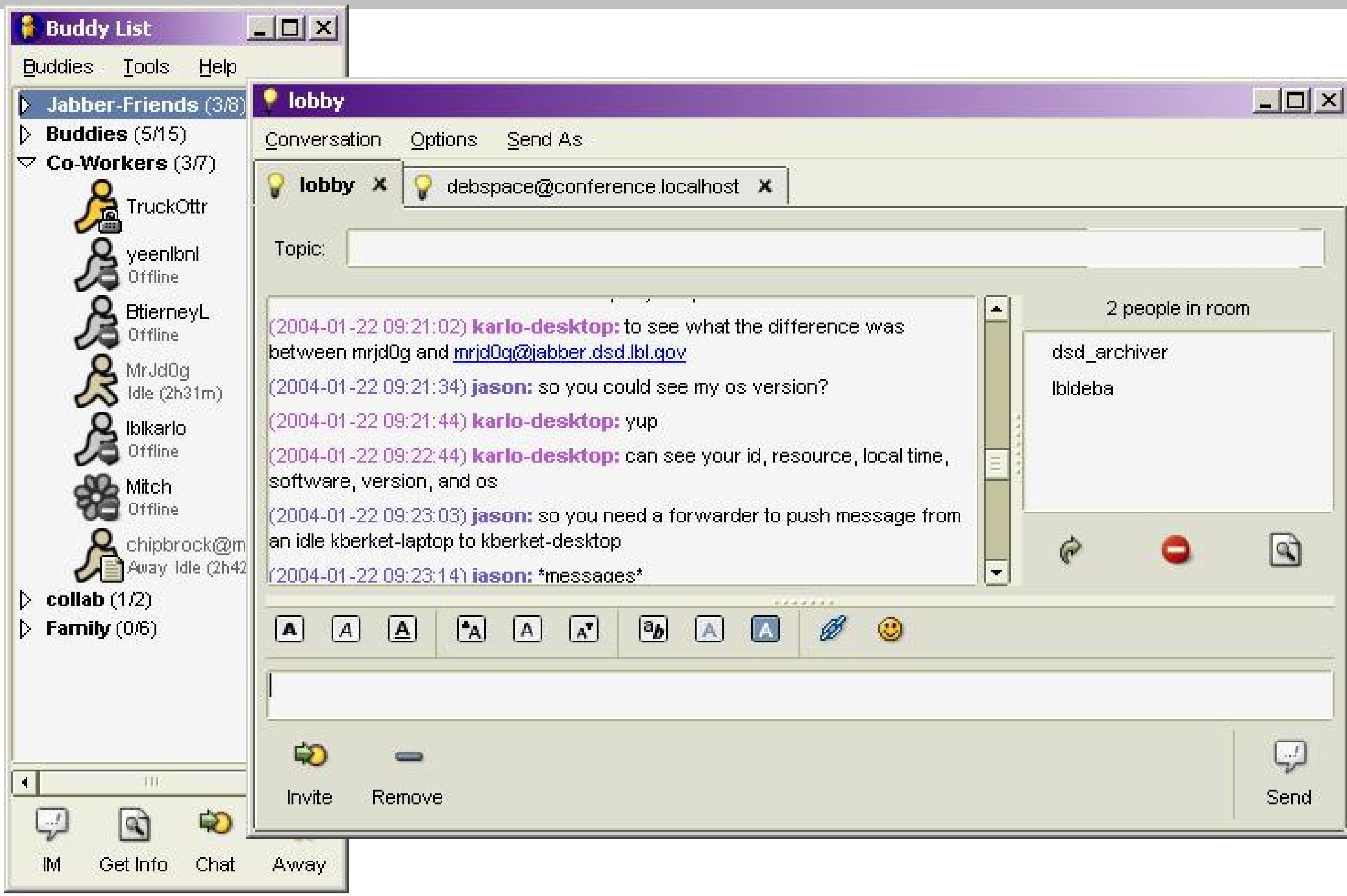
Pervasive Collaborative Computing Environment Goals

- Support ‘continuous’ collaboration
  - Ubiquitous – available anywhere
  - Synchronous and asynchronous
  - Persistent
- Low threshold for entry into the environment
- Target daily tasks and base connectivity
- Leverage off of existing components
- Secure environment
- Scale to support small and large groups

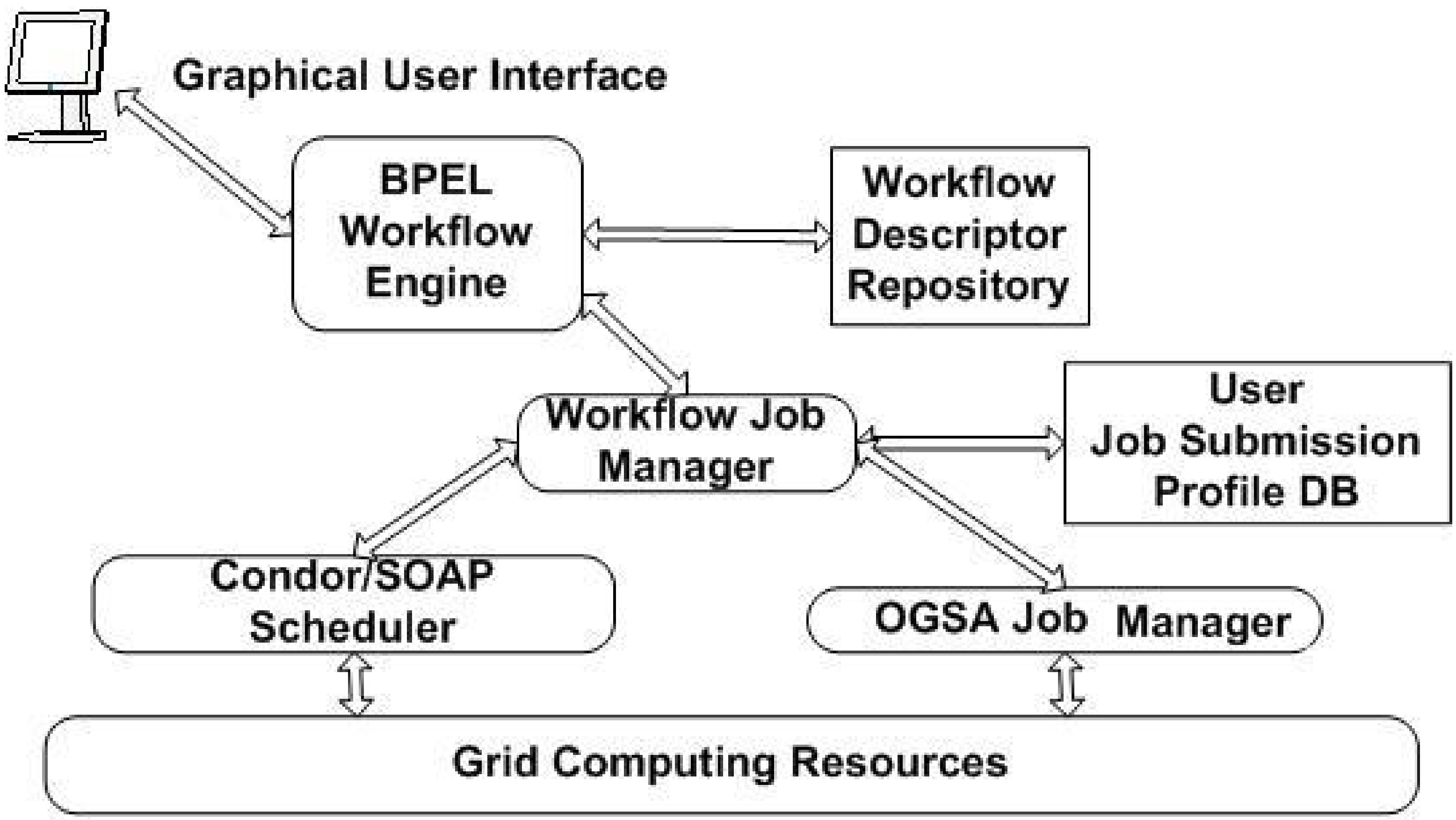
U.S. Department of Energy

Office of Science

PCCE – XMPP-based Secure Messaging



Collaborative Workflow



Abstract Workflow Job Manager

Scalable and Secure P2P Information Sharing

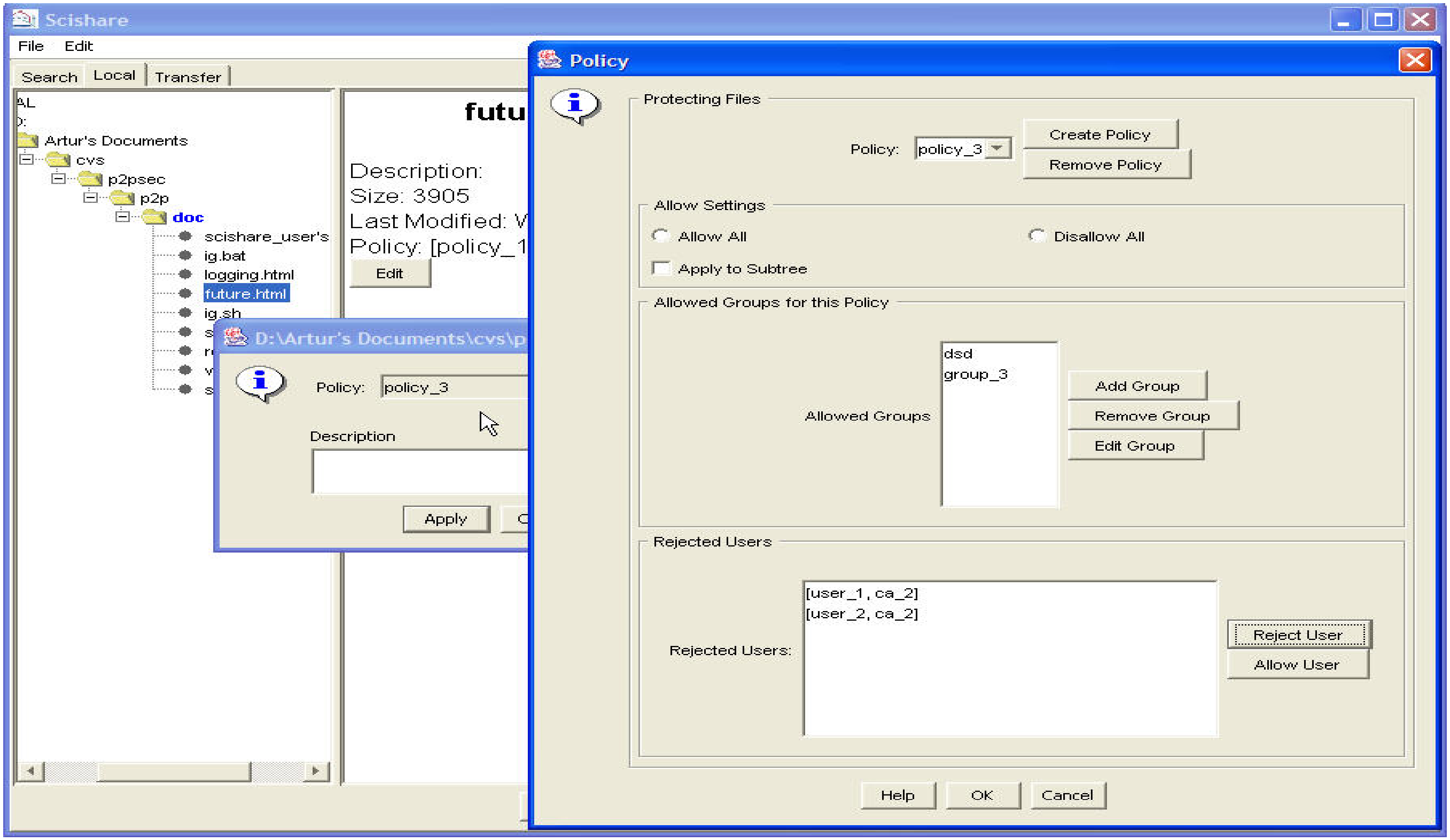
- Create a peer-to-peer system to support location independent information sharing in the scientific community
- Goals
  - Security (flexible)
  - Scalability (group communication)

U.S. Department of Energy

Office of Science

GUI

BERKELEY LAB



Security

- Goals
  - Identify users – authentication
  - Define and enforce access control – authorization
  - Protect confidentiality of data – encryption
  - Define roles and levels of trust
  - Easy to configure and use from any location
- Tools
  - Akenti authorization server
  - Secure group layer
  - Message level security
  - Incremental trust

Akenti Goals

- Access based on policy statements made by multiple independent stakeholders
- Use Public Key Infrastructure (X.509) standards
  - To identify users
  - Create digitally signed certificates
  - Use TLS/GSI authenticated connections
- Targeted at distributed environments
  - Users, resources, stakeholders are geographically and administratively distributed

Akenti Policy

- Minimal local authorization policy files:
  - Who to trust, where to look for certificates.
- Most access control policy contained in distributed digitally signed certificates:
  - X.509 certificates for user identity and authentication
  - UseCondition certificates containing stakeholder policy
  - Attribute certificates in which a trusted party attests that a user possesses some attribute, e.g. training, group membership

Python CoG Kit

- Provide a mapping between Python and the Globus Toolkit®.
  - Clean object-oriented interface
  - Similar performance to the underlying C code
  - Stability between versions of the Globus toolkit
  - Natural to use interface
- Available for Globus 2.2.4 or 2.4
- Distributed as part of the Globus 3.2 release

Grid Services Project

- Develop Open Grid Services Architecture implementation in Python
  - Grid Services Specifications
  - Open Grid Services Infrastructure
    - Standalone implementation in python
      - Both client and server side
  - Higher-level Services
  - Application-specific services and infrastructure
- Recent WS-RF changes will be tracked

DOE Science Grid

- Support DOE's large scale science projects by providing cyber infrastructure that is persistent, scalable, and community standards-based
  - LBNL, ANL, ORNL, and PNNL
  - NERSC
  - Provide needed tools
  - Perform outreach to applications
  - Reduce barriers to use

End-2-End Monitoring

- Improve end-to-end data throughput for data intensive applications
- Provide the ability to do performance analysis and fault detection
- Provide accurate, detailed, and adaptive monitoring of all of distributed computing components, including the network
- This requires a unified view of a wide range of sensor data, from network to host to application

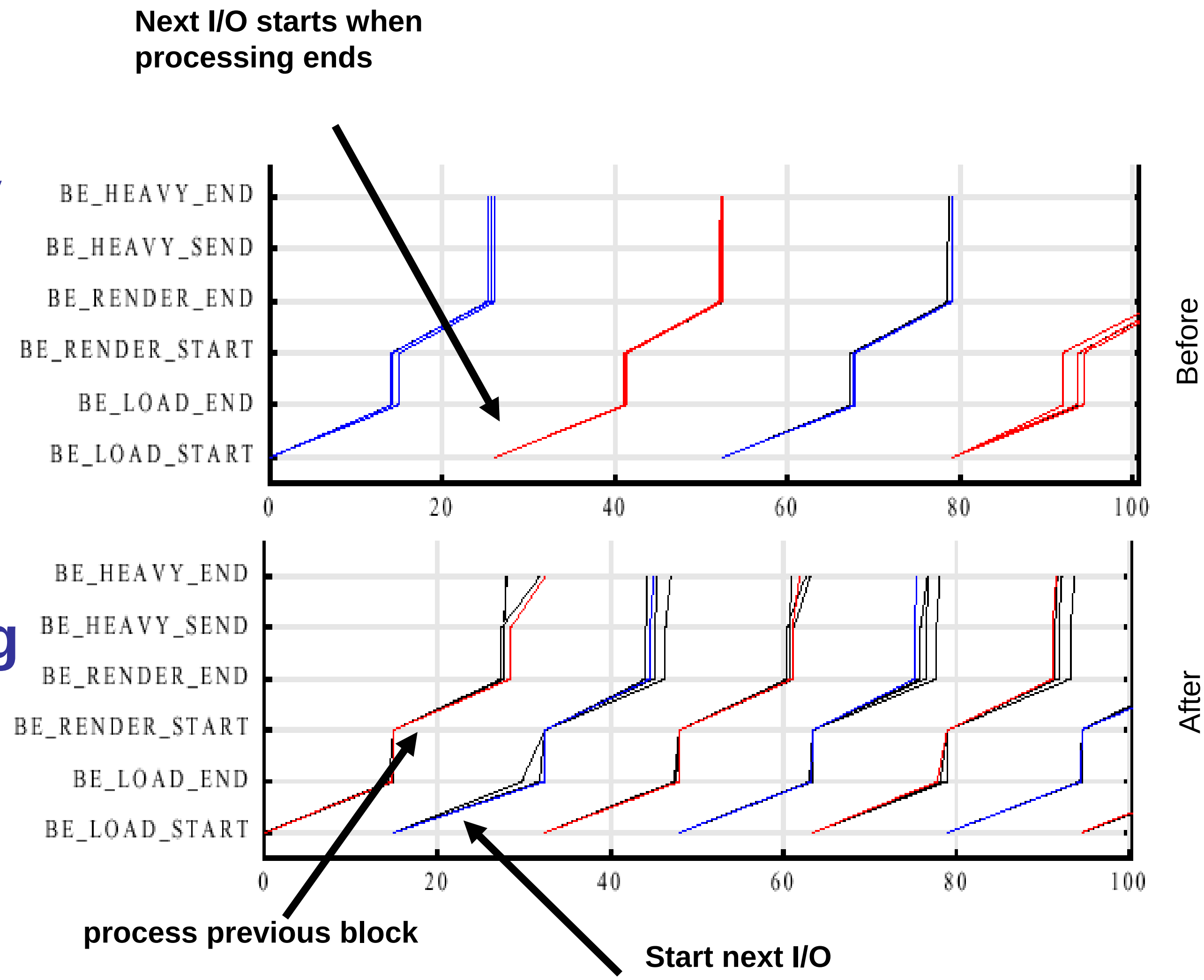
NetLogger Toolkit

- The NetLogger Toolkit includes:
  - Tools to make it easy for distributed applications to log interesting events at every critical point
    - NetLogger client library (C, C++, Java, Perl, Python)
    - Extremely light-weight: can generate > 900,000 events / second on current systems
  - Tools for host and network monitoring
  - Event visualization tools that allow one to correlate events
  - NetLogger event archive and retrieval tools
- NetLogger can provide a complete view of the entire system.

Detailed Application Instrumentation Example

- **I/O followed by processing**
- **overlapped I/O and processing**

almost a 2:1 speedup



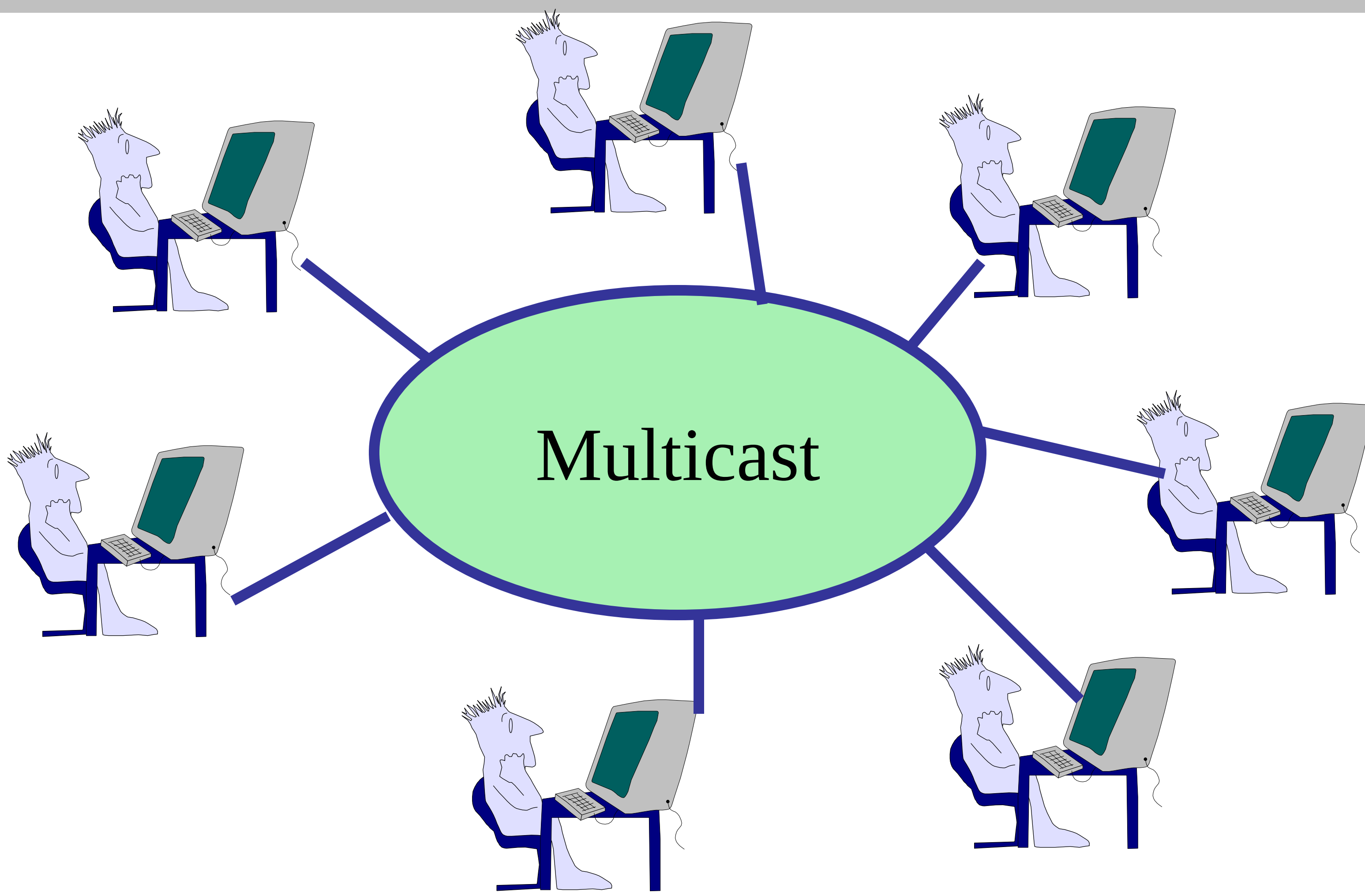
Improving TCP Performance

- Goal - improve performance on high bandwidth*delay product links
- TCP Tuning Guide
- Web100/Net100 instrumented TCP and WAD
- Simulate HighSpeed TCP (proposed by Sally Floyd)
  - Bulk transfer capabilities
  - Fairness
  - Response to active queue management

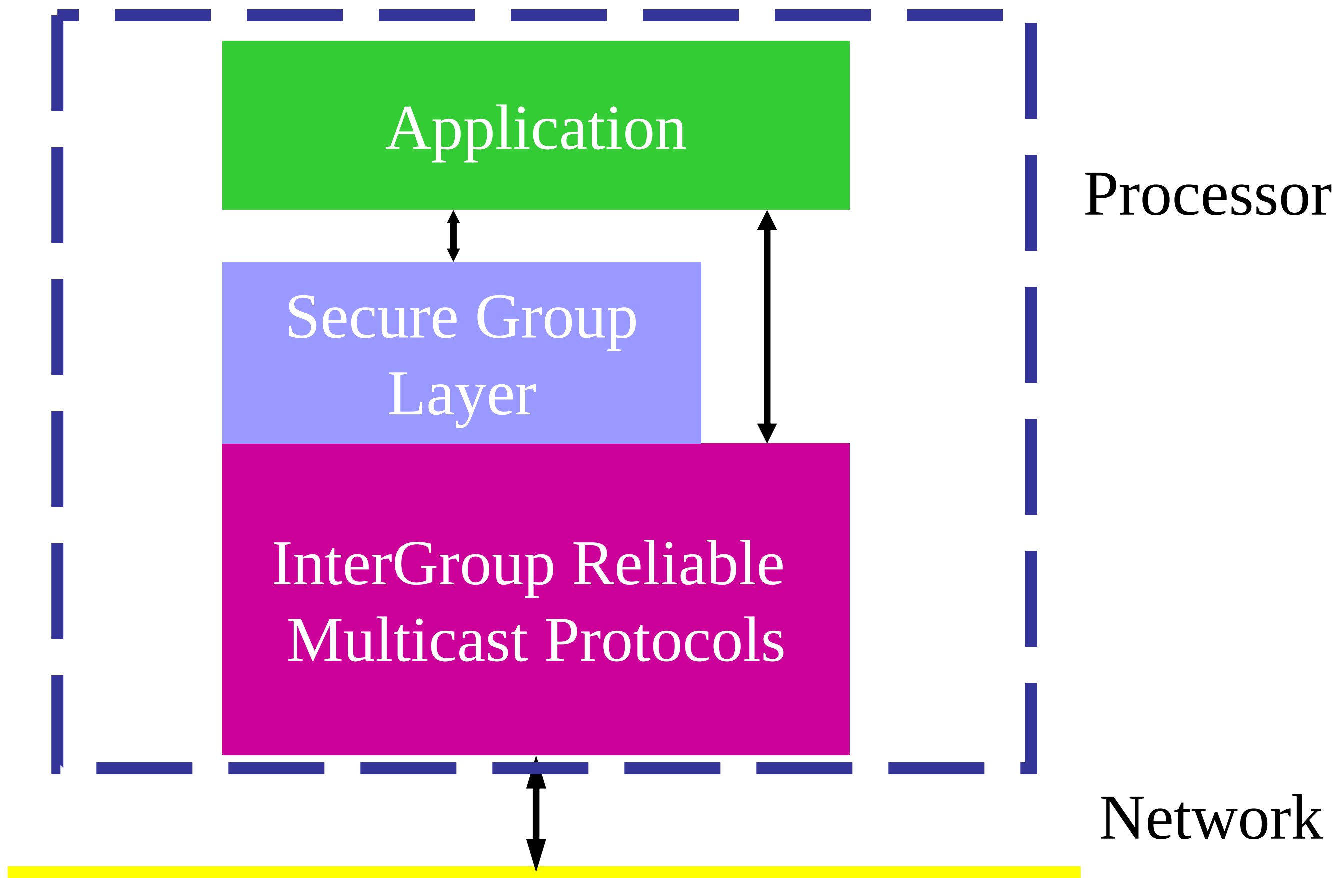
Communication Protocols to Support Collaboratories

- Scalability
  - High data throughput
  - Large numbers of users interacting
- Support for peer-to-peer communication
- Robust – not dependent on servers

Group Communication



InterGroup + SGL



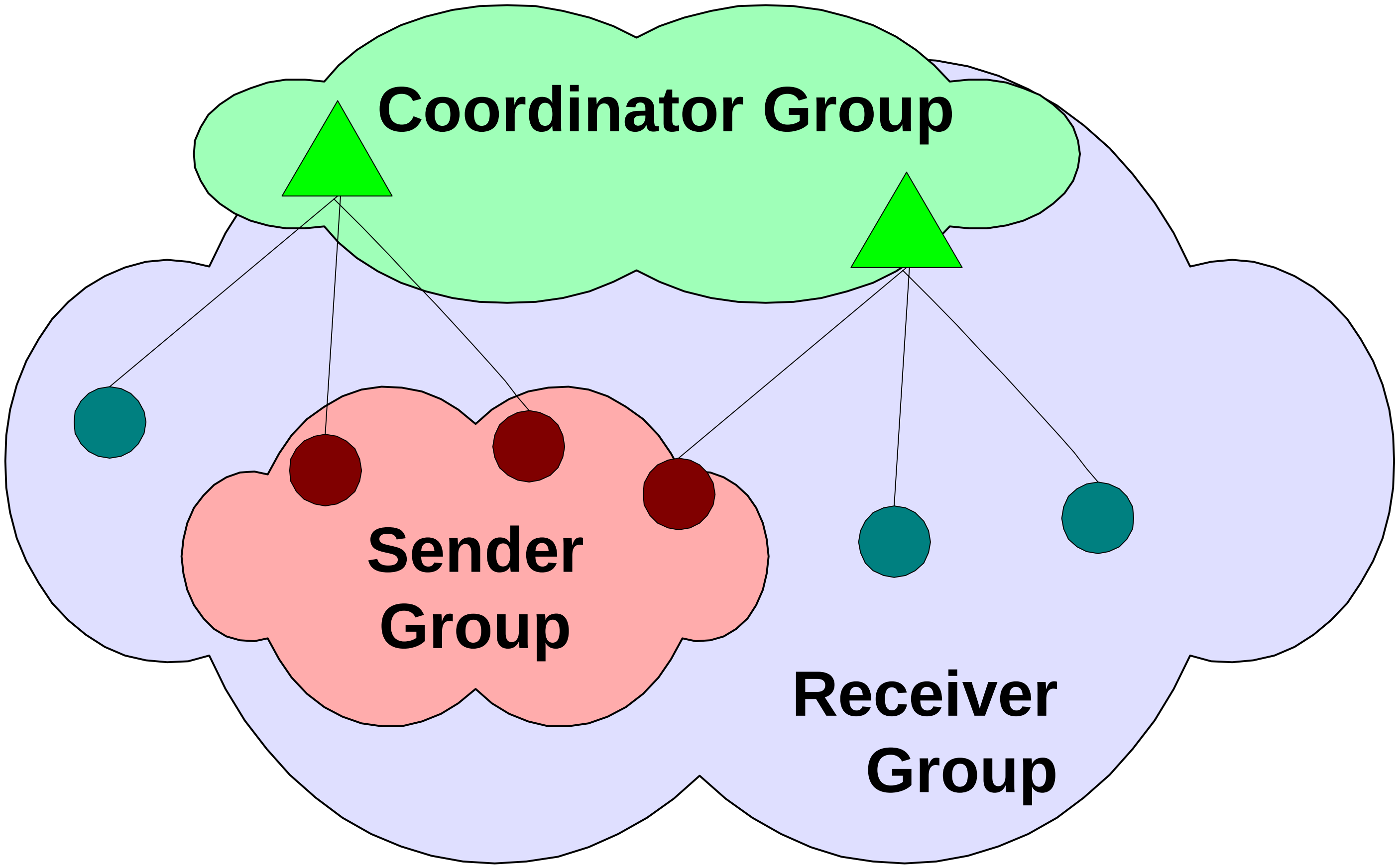
InterGroup Protocols

- Goals
  - – Support a broad range of applications
    - Broadcast – one-to-many
    - Many-to-many
  - – Provide a broad range of guarantees
    - Reliable and unreliable delivery
    - Sender order, total order, and unordered
  - – Based on IP Multicast
- Scale to the Internet
  - – Many groups
  - – Many members in each group
  - – Heterogeneous latency between members

InterGroup Protocol

- All members of the group can send messages to the group
- All processes in the group receive the messages sent within the group
- Membership tracking with notification of membership changes
- Messages delivered at each member of the group in a consistent order
  - total order (timestamp)
  - preserve causality
  - membership changes ordered with respect to messages

InterGroup Schematic

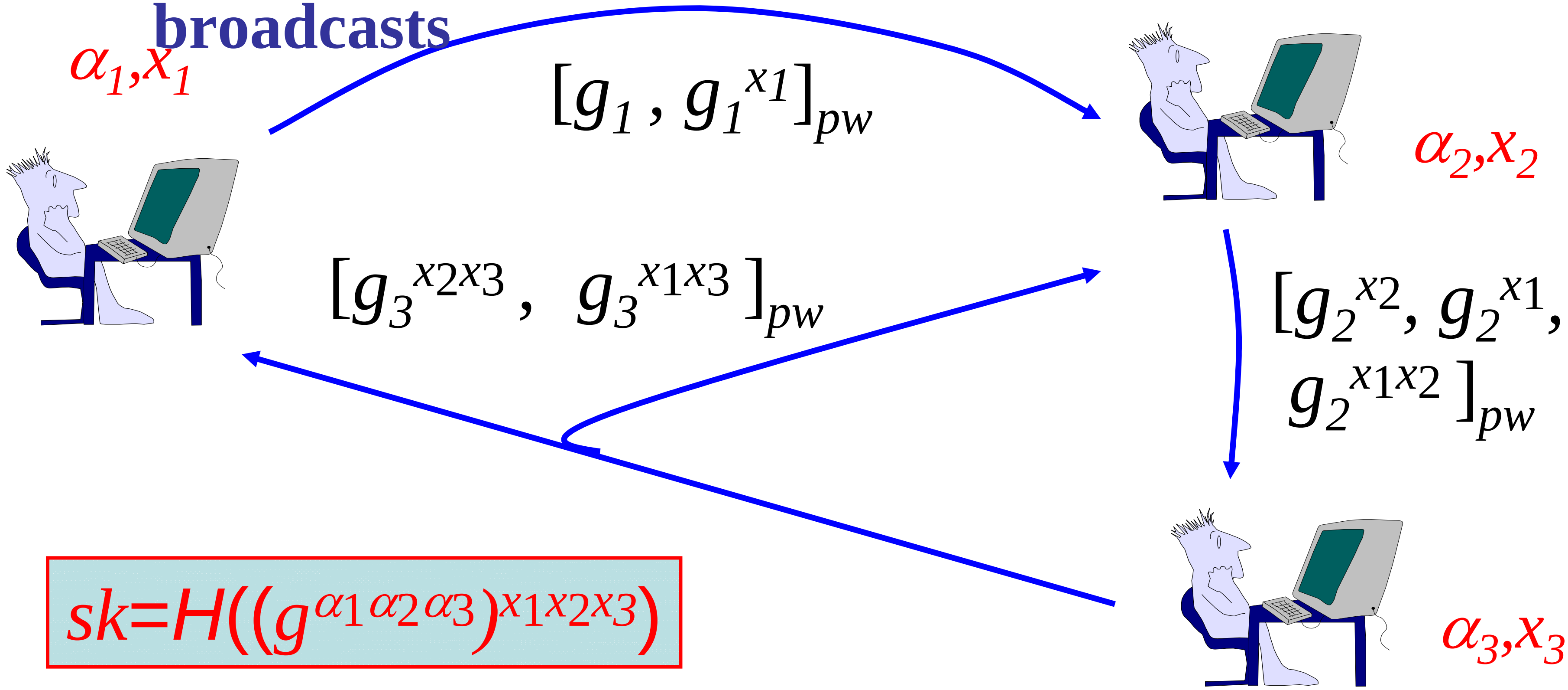


The Secure Group Layer: SGL

- A group Diffie-Hellman key exchange algorithm enables group members to establish a session key
- Symmetric crypto algorithms (e.g. DES and HMAC)
  - implement a secure channel
- An access control mechanism makes sure that only the legitimate parties have access to the session key
  - certificate-based
  - password-based
- Provable security

Group Diffie-Hellman Algorithm

- **Up-flow: U_i raises received values to the power of the values (x_i, α_i) and forwards to U_{i+1}**
- **Down-flow: U_n processes the last up-flow and broadcasts**

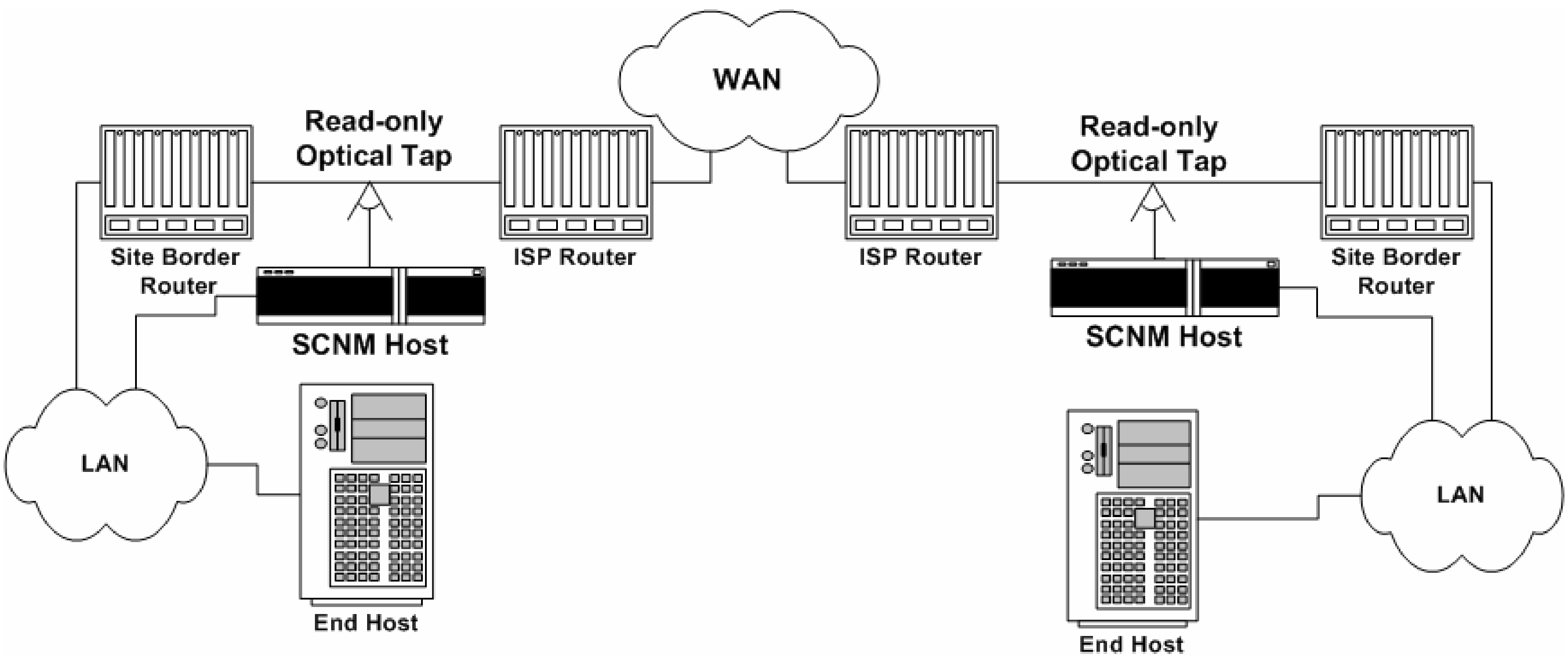


$\boxed{sk = H((g^{\alpha_1 \alpha_2 \alpha_3})^{x_1 x_2 x_3})}$

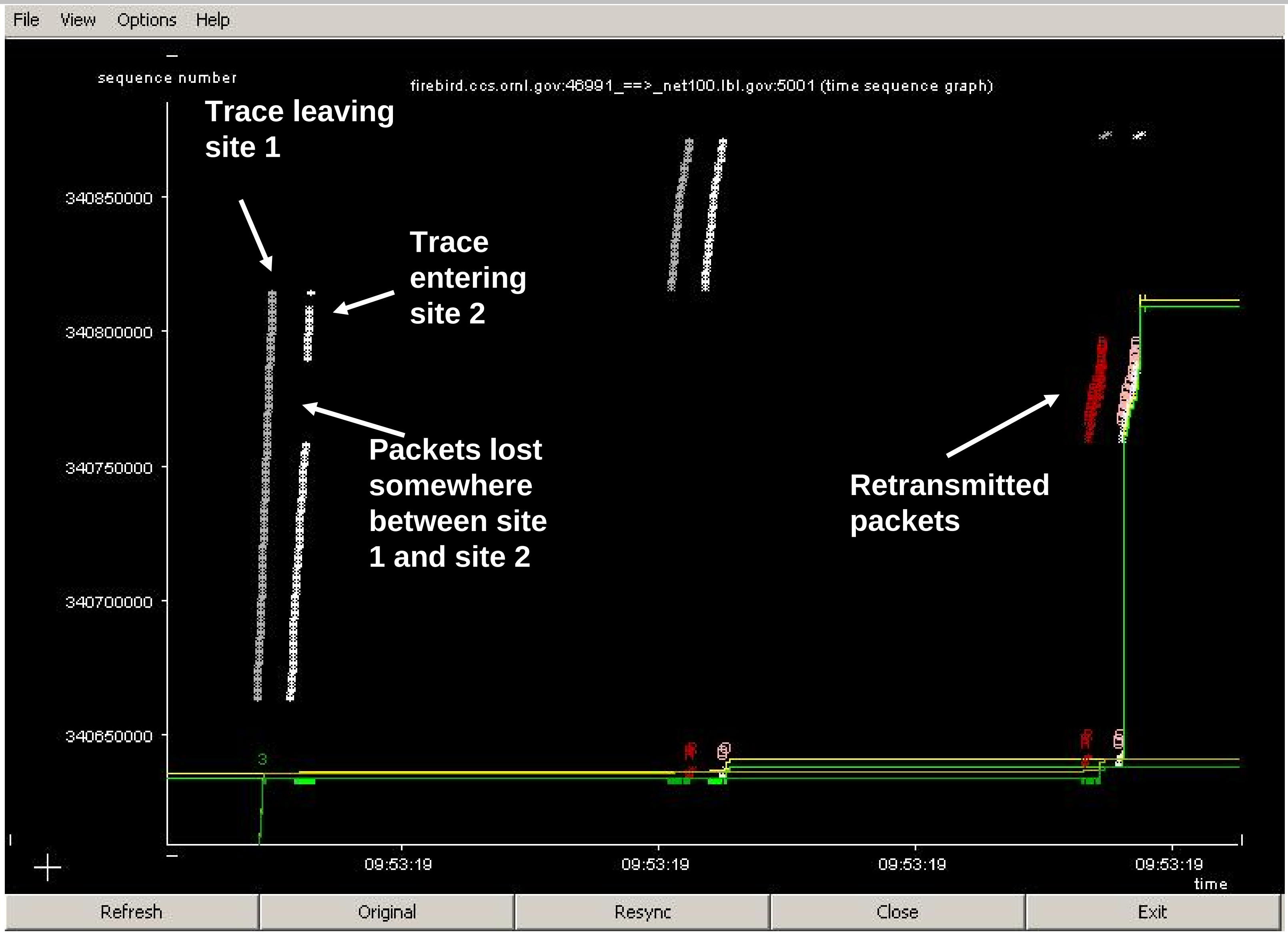
Self Configuring Network Monitor (SCNM)

- **SCNM is a passive monitoring system designed to address the following issues:**
  - **Ability for network *users* to monitor their own traffic**
  - **Ability to identify the source of network congestion or other problems (e.g: a LAN or WAN issue)**
  - **Ability for application developers to characterize their own traffic, and how it is impacted by the network**
  - **Protocol analysis and debugging**
    - **Often not possible to capture packet traces at the sending host**
      - **tcpdump will often lose packets when trying to capture a high bandwidth stream**

Step 5: Add Passive Monitoring "Inside" the Network



Typical Passive Header Capture Results



URL

- http://www.dsd.lbl.gov/